



Universitat de Lleida

TREBALL FINAL DE GRAU



ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA
INSPIRING THE FUTURE

Estudiant: Carlos Isaac Parra

Titulació: Grau en Tècniques d'Interacció Digital i Computació

Títol de Treball Final de Grau: Aplicación móvil de moda para la fidelización del cliente

Director/a: Jordi Mateo Fornés

Presentació

Mes: Setembre

Any: 2021

Índice de contenido

Abstract	4
1. Introducción	5
2. Objetivos	5
3. Metodología	5
3.1 Tecnologías seleccionadas	5
3.2 Planificación del proyecto.....	6
3.3 Seguimiento del proyecto.....	7
4. Análisis de Requisitos	8
4.1 Requisitos software	10
4.2 Requisitos de usuario	10
5.Implementación	10
6. Resultados	16
6.1 Funcionalidades y producto final.	16
7. Conclusiones	22
8. Bibliografía.....	23

Índice de figuras

Fig. 1: Cargar APP de prueba.....	8
Fig. 2: Menú APP de prueba	8
Fig. 3: Tiendas APP de prueba	9
Fig. 4: Opciones APP de prueba	9
Fig. 5: Menú “Bottom Bar” Pull and Bear	11
Fig. 6: Artículos Pull and Bear	11
Fig. 7: Tiendas Bershka.....	12
Fig. 8: Mapa de tienda Bershka	12
Fig. 9: Toggle Switch editado	13
Fig. 10: Toggle Switch predeterminado	13
Fig. 11: MVC.....	13
Fig. 12: Factory Bean diagrama UML	14
Fig. 13: Patrón Singleton UML	14
Fig. 14: Base de datos APP.....	15
Fig. 15: Obtener artículos con RemoteObjetct	15
Fig. 16: Cargar APP.....	16
Fig. 17: Artículos APP	17
Fig. 18: Detalle artículo APP	17
Fig. 19: Usuario APP	18
Fig. 20: Cerrar sesión APP.....	18
Fig. 21: Iniciar/Registrar usuario APP	19
Fig. 22: Tiendas APP	20

Fig. 23: Detalle tienda APP.....	20
Fig. 24: Mapa de Google de Tienda	21
Fig. 25: Configuración APP	22
Fig. 26: Errores APP.....	22

Índice de tablas

Tabla 1: Planificación Octubre-Enero.....	7
Tabla 2: Planificación Febrero-Junio	7

Abstract

El proyecto consiste en crear una aplicación móvil para Android y iOS con el objetivo de fidelizar a los clientes con mensajes, promociones, ofertas, recomendaciones... Esta aplicación se creó dentro de la empresa que trabaja el estudiante, Intarex. Esta se especializa en el desarrollo de soluciones verticales para empresas textiles de moda y calzado, curtidurías y ópticas, integradas mediante el ERP SAP Business One.

Este proyecto lo plantearon hace un tiempo, pero no le dieron prioridad, ahora han visto la mejor oportunidad para realizarlo, gracias a la entrada del estudiante a la empresa y para organizar mejor las empresas que gestionan ya que tienen mucho trabajo, así que, con ello, el estudiante podrá demostrar sus capacidades tanto de desarrollador como de analista y mejorar la interfaz de usuario para llevar este proyecto a buen puerto.

1. Introducción

Este proyecto se ha llevado a cabo mientras Carlos Isaac Parra, estudiante de 3o del Grado en Técnicas de Interacción Digital y de Computación, realizaba en la empresa Intarex, S.L. la Formación Dual, que consiste en la alternancia de los estudios de los aprendizajes en un entorno académico y un entorno profesional, en este caso, entre universidad y empresa, donde el estudiante se convierte en el aprendiz que estudia. Y por ello las empresas se implican en la Formación Dual de la Universidad de Lleida en el reclutamiento de los/las aprendices, tener una evaluación periódica del aprendiz, hacer reuniones de trabajo, prospectivas o de evaluación, centradas en la progresión del aprendiz y dar enseñanzas tutorizadas por profesionales.

La Empresa

Intarex, S.L. es una empresa igualadina que se especializa desde hace más de 25 años en el desarrollo de soluciones verticales para empresas textiles de moda y calzado, curtidurías y ópticas, integradas mediante el ERP SAP Business One que tiene como objetivo que el cliente desarrolle todo su potencial. En la empresa hay más de 30 profesionales, donde se incluyen licenciados en informática o matemáticas, técnicos en electrónica y másteres en gerencia empresarial. A raíz de la pandemia mundial la empresa ha realizado soluciones tecnológicas para resolver las problemáticas de sus clientes, y ha ayudado a la transición al teletrabajo de muchos de sus clientes como por ejemplo hacia el “Cloud de Azure” para tener todos los documentos, programas y correos electrónicos en los servidores de la nube para acceder por internet.

Como se ha comentado en el abstract, la empresa tiene la necesidad de retomar el proyecto de aplicación de fidelización de sus clientes, donde se partirá de una pequeña aplicación que creó la empresa hace tiempo, pero que no fue llevada a cabo por falta de experiencia en temas de diseño orientado al usuario, aplicaciones con menú bottom bar, que son unos de sus requisitos.

La empresa tiene como objetivos avanzar el máximo trabajo del proyecto posible con la incorporación del estudiante, así como conseguir la construcción y adquisición de nuevos conocimientos y de las competencias a tratar para el futuro del estudiante dentro de la empresa.

Con la primera versión de la APP, se comenzó a hacerle una comparativa con otras aplicaciones del estilo, con ello, se adquirió el conocimiento para poder crear una app de 0, utilizando los nuevos requisitos, siguiendo los patrones de una APP de Android. Las aplicaciones con las que se comparó se utilizan para tener acceso a la tienda, para comprar ropa online, descubrir las últimas novedades, tendencias... Entre ellas se compararon:

- Pull and Bear, donde tenía en común el apartado de listar artículos, donde se mostraba más información, una foto del artículo, el apartado de usuario.
- Bershka, de esta se comparó el apartado de tiendas, donde se muestra información de Google, a parte de la del contacto a la tienda.

2. Objetivos

- Desarrollo e Implementación de una aplicación multiplataforma.
- Integración de la app en el workflow de la empresa.
- Análisi del impacto del uso de la app.
- Análisis y mejora de la interfaz de usuario.
- Conocer la API de SAP para la gestión de datos.
- Aprender y saber utilizar la parte de desarrollo de aplicación con flex.

3. Metodología

3.1 Tecnologías seleccionadas

Las tecnologías y lenguajes utilizados para realizar el proyecto han sido:

-**IDE de Eclipse**, es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma, aunque típicamente se ha utilizado para desarrollar otros entornos de desarrollo integrados, también se puede usar para crear proyectos de aplicaciones (móviles o de escritorio). Este IDE dispone de un editor de texto con analizador sintáctico, compilación en tiempo real, se pueden utilizar el control de

versiones a partir de CVS y a través de “plugins” se añaden diferentes funcionalidades para poder soportar más idiomas.[1]

Este IDE fue utilizado porque es el que utiliza la empresa para crear aplicaciones, y es más sencillo de conectar la parte servidor y cliente de los demás IDEs que he utilizado.

-**Apache Flex**, anteriormente llamado Adobe Flex, es un kit de desarrollo de software (SDK) para el desarrollo y la implementación de aplicaciones web enriquecidas multiplataforma basadas en la plataforma Adobe Flash, publicado inicialmente en marzo de 2004 por Macromedia, luego adquirido por Adobe Systems. Las aplicaciones que utilizan estas tecnologías se pueden desarrollar mediante entornos de desarrollo integrados (IDE) estándar (en mi caso, Eclipse). Se compilaban a través del lenguaje de marcas Flex (**MXML**) y se ejecutaban mediante **ActionScript** aplicaciones Flash.

Algunas características del proceso de desarrollo de una aplicación Flex son definir una interfaz de aplicación usando un conjunto de componentes Spark (textos, botones, listas...), ordenar estos componentes en el diseño de la interfaz de usuario, usar y editar estilos y temas para definir el diseño visual, añadir comportamiento dinámico, definir y conectar a servicios de datos (imágenes, artículos...) según sea necesario, como con servicios HTTP. [2]

-**MXML (Macromedia eXtensible Markup Language)**, es un lenguaje basado en XML(metalenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible), que describe interfaces de usuario, crea modelos de datos y tiene acceso a los recursos del servidor, del tipo RIA (Rich Internet Application).[3]

-**ActionScript**, es un lenguaje de programación que crea un conjunto de instrucciones de un programa. Este lenguaje crea script en Flash, con el que crear películas con elementos interactivos.[4]

Hemos utilizado este SDK ya que la empresa lo ha utilizado siempre, y aunque exista poca información en internet por el poco uso que se le da actualmente, se pudo aprender mediante algunas clases que dio la empresa.

-**SAP**, es un sistema de planificación de recursos empresariales (ERP) que integran y manejan muchos negocios que tienen relación con la producción y distribución de bienes y servicios.[5]

Como la aplicación está destinada a los clientes de mi empresa (mayoritariamente empresas de moda), utilizamos este ERP con las bases de datos necesarias para implementar mi aplicación y también aprender a manejar los datos con algunas clases de mi empresa por si hubiese alguna necesidad de cambiarlos.

-**SAP HANA**, es una base de datos en memoria de alto rendimiento, con el objetivo de ofrecer un mejor rendimiento de las aplicaciones analíticas y transaccionales (tratamiento de transacciones online). [6]

Como se usó el ERP de SAP y suele utilizar HANA para las consultas, esta tecnología tiene algunas reglas en el momento de crear las consultas SQL a su BBDD que si no se cumplen puede haber fallos en la aplicación, así que, para no entrar en detalle en consultas concretas de cada tecnología, se hicieron consultas que todas soportan, y se trataron los datos desde Java.

3.2 Planificación del proyecto

<i>Planning</i>	2020-2021															
	OCT				NOV				DIC				ENE			
Semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Actividad 1: Aprender y saber utilizar la parte de desarrollo de aplicación con flex (parte cliente)																
Análisis y mejora de la interfaz de usuario																
Generación de la documentación																
Actividad 2: Desarrollo del servidor																
Comunicación con servidor de la parte cliente (Flex)																

[illegible]

Tabla 1: Planificación Octubre-Enero

<i>Planning</i>	2020-2021																			
	FEB				MAR				ABR				MAY				JUN			
Semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Actividad 1: Continuar aprendizaje durante el desarrollo de la aplicación con flex (parte cliente)																				
Análisis y mejora de la interfaz de usuario																				
Actividad 2: Desarrollo del servidor																				
Comunicación con servidor de la parte cliente (Flex)																				
Conocer la API de SAP para la gestión de datos																				
Conocer el marco empresarial para empresas del sector de la moda																				

Tabla 2: Planificación Febrero-Junio

3.3 Seguimiento del proyecto

Para hacer el seguimiento del proyecto se realizaron una serie de reuniones a lo largo del curso, donde se revisaban si se habían cumplido los requisitos o tareas pedidas por la empresa, y se buscaban nuevos objetivos a cumplir, muchas veces estos no tenían una fecha específica de finalización, ya que la empresa no sabía cuánto se podía tardar en realizar por la inexperiencia en este tipo de aplicaciones.

Primeras reuniones

Para comenzar el proyecto hubo algunas clases de formación presenciales en la empresa donde se hizo una presentación al lenguaje de Flex mediante el IDE Eclipse Juno. Después de ello, para iniciarse en el entorno se hizo una aplicación de prueba para moverse un poco por el entorno, y aprender el lenguaje, esta aplicación tiene funcionalidades básicas como crear pantallas, con botones que hacían algunas acciones como avanzar de pantalla, ir para atrás, hacer contadores, me ayudé para hacerlo con la página facilitada de Tour Flex, actualmente no se puede acceder dado que Adobe ya no admite Flash Player. Con ello, se pudo proseguir con el apartado de conexión del servidor con la APP, con nuevas clases en la empresa, cuando se consiguió la conexión con la aplicación de ejemplo, se pudo hacer una reunión para empezar con el proyecto final.

Reuniones de desarrollo de proyecto

Esta parte se dedicó a reuniones menos constantes ya que al tener una base de lo que se quería hacer, se pudo trabajar por objetivos de la aplicación. Teniendo un registro diario de lo que se había avanzado en la APP, donde los trabajadores de la empresa podían mirar si se estaban obteniendo los resultados adecuados.

Reuniones de pruebas

Quando se tuvo una demo funcional acabada de la aplicación, se realizó una reunión donde se mostró a los empleados de la empresa la APP para que dieran su opinión, qué puntos se deberían revisar... Y más adelante tener otra reunión comprobando que se hayan realizado los nuevos requisitos.

4. Análisis de Requisitos

En este apartado la empresa tuvo un papel importante ya que tenía una base definida de la anterior app que intentaron realizar y pudieron proporcionar sus conocimientos, y así tener una idea de lo que nuestro usuario espera de la aplicación, obteniendo así los requisitos del usuario y del software.

Esta APP tenía algunos apartados no funcionales, inacabados, no seguían ningún diseño y no cumplían con los requisitos de la empresa. Los fallos que se encontraron fueron:

- Apartado de cargar la APP *Fig.1*: no te daba la información de que app estabas abriendo, si el servidor no obtenía datos, se quedaba en la misma pantalla y no se podría saber si está cargando o no funciona.

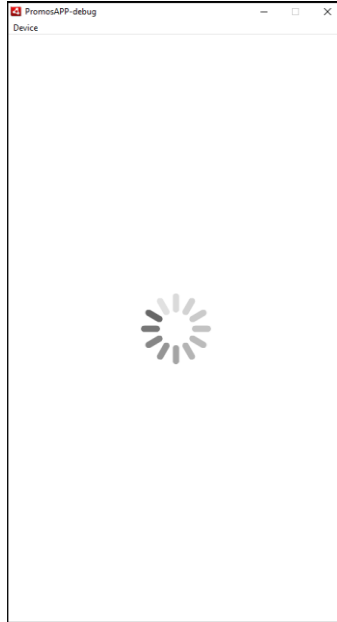


Fig. 1: Cargar APP de prueba

- Menú inicial *Fig.2*: la empresa pedía que se abriese una pantalla principal con catálogo y que tuviera un menú “bottom bar navigation”.

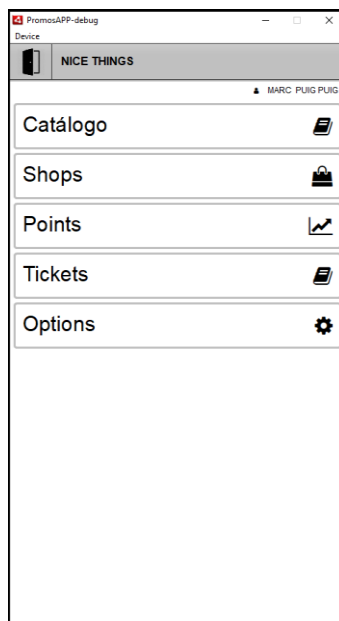


Fig. 2: Menú APP de prueba

- Las listas de tiendas, artículos, vales *Fig.3*: No daban la información suficiente, con tan solo el nombre o el código de estos, el diseño era pobre, ya que utilizaba los objetos base de Flex, con colores apagados, lo que hace que no llame la atención del usuario.

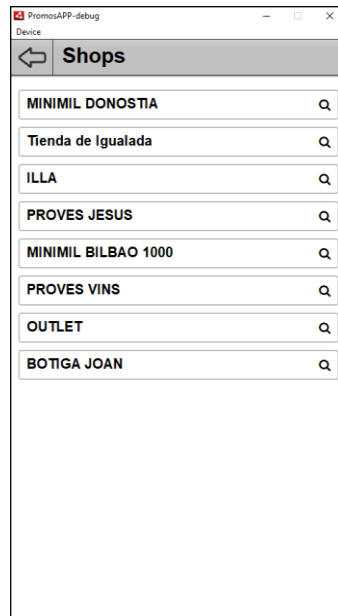


Fig. 3: Tiendas APP de prueba

- Opciones *Fig.4*: Utilizaba otro estilo diferente de botón para ir atrás, que suele ser incorrecto en aplicaciones. La pantalla no se aprovechaba, se podía añadir más información y funcionalidades, como la versión de la APP.

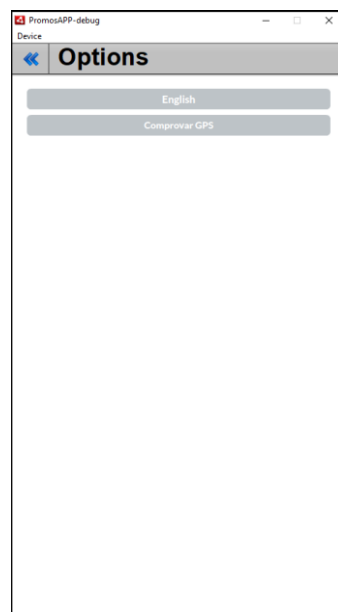


Fig. 4: Opciones APP de prueba

Debido a los aspectos comentados, se tomó la decisión de que sería más sencillo empezar la aplicación desde cero que continuar con ella.

Los requisitos del usuario y del software detectados fueron:

4.1 Requisitos software

Los requisitos software de la APP definen el comportamiento del sistema que se va a desarrollar. Como casos de uso (requisitos funcionales) de todas las interacciones que tendrán los usuarios con el software. En este caso tenemos:

Requisitos funcionales

- 1- Acceso al teclado del móvil.
- 2- Recoger datos de la base de datos de SAP.
- 3- Un menú “bottom bar navigation” para acceder a las diferentes funcionalidades.
- 4- Editar el tema de la app a oscuro o claro.
- 5- Filtros de la ropa activados o no.
- 6- Mostrar código de barras de los productos.
- 7- Filtrar artículos por color.
- 8- Cuando los usuarios quieran registrarse, se les redirigirá a una web, donde tendrán que rellenar un formulario. Este se quedará guardado en la base de datos de SAP.
- 9- Cambio de idioma de la aplicación
- 10- Acceso a las notificaciones del dispositivo.
- 11- Roll usuario registrado o no registrado.

Requisitos no funcionales

- 12- Base intermedia de MySQL para guardar datos esenciales para el funcionamiento de la aplicación, por si SAP dejase de funcionar.
- 13- Servidor que deje un servicio abierto de nuestra aplicación para que pueda ser utilizada con datos actualizados en el momento.
- 14- Descargar imágenes al iniciar el servidor, para poder obtenerlas sin hacer peticiones a la base de datos. Por ello, todas las imágenes deberán estar organizadas por igual para cualquier tipo de producto (carpeta con el nombre del producto, imagen con el mismo nombre y para especificar colores nombre + código de color).
- 15- Web de Test para funcionalidades del servidor. (devolver parámetros, descargar imágenes...)

4.2 Requisitos de usuario

Los requisitos de usuario son las expectativas que se tiene sobre el sistema para cumplir con sus necesidades, con ellos se logrará conseguir la fidelización del cliente, ya que son funcionalidades que harán que tenga un registro de su usuario en las diferentes tiendas en las que esté, obtener los datos de las tiendas y utilizar vales de descuento, así que se obtienen:

Requisitos no funcionales

- 1- Mostrar catálogo de ropa
- 2- Filtros para el catálogo
- 3- Filtros para el detalle de la ropa
- 4- Darse de alta en la aplicación
- 5- Iniciar/Cerrar sesión
- 6- Configuración de la aplicación
- 7- Tiendas cercanas al usuario
- 8- Perfil de usuario
- 9- Vales de descuento del usuario
- 10- Compras recientes de usuario

5.Implementación

Implementación de la APP

Una de las primeras fases que se tuvo en la implementación de la APP fue la de diseño, donde la empresa me mostró la aplicación que dejaron de lado, donde era difícil de usar, el diseño era pobre, y al no seguir prácticamente ningún principio de diseño la confianza con la aplicación será muy baja y no se descargará(visto en el apartado de análisis de requisitos). Así que me encargué de realizar una investigación de cómo diseñar en Flex por mi

cuenta, donde a pesar de que había poca información y ejemplos, pude sacarlo adelante, lo que se hizo primero fue comparar con otras aplicaciones del estilo para decidir qué tipo de interfaz se adecuaba mejor para el usuario, estas fueron:

-Pull and Bear, aplicación donde se pudo sacar una idea del menú “bottom bar” Fig.5 ya que se creyó buena idea mostrar un icono junto a una palabra clave del apartado, también se tomó una idea de cómo está organizada y cómo tiene distribuida la información de las listas de artículos porque daba la información necesaria para el usuario, dando más importancia al aspecto visual.

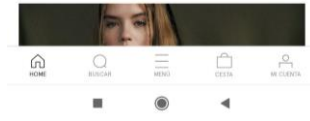


Fig. 5: Menú “Bottom Bar” Pull and Bear

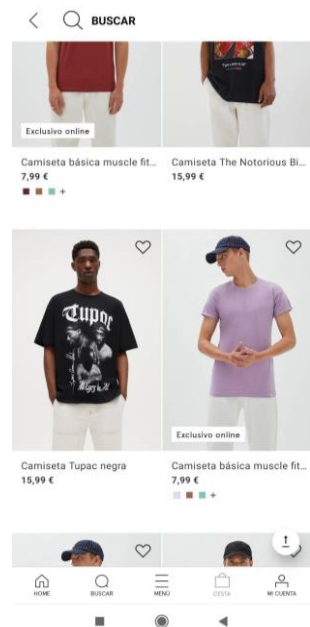


Fig. 6: Artículos Pull and Bear

-Bershka, en esta aplicación se tomó como ejemplo la manera en que tenían organizadas sus tiendas, ya que da la información necesaria en la lista de tiendas Fig.7 (nombre, lugar, distancia) y cuando se accede a una en concreto te da información más específica, y un mapa de Google Fig.8, donde la empresa creyó más acertado redirigir a través de un botón a Google Maps con la ubicación exacta de la tienda, y desde este punto poder ir a la ubicación de la tienda o acceder a más datos de ella.

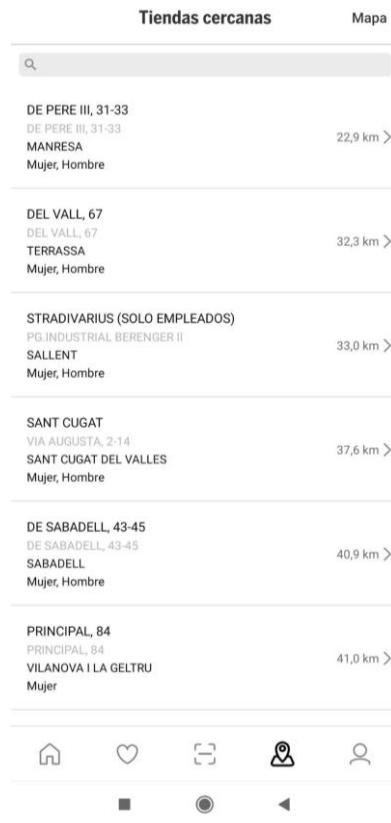


Fig. 7: Tiendas Bershka

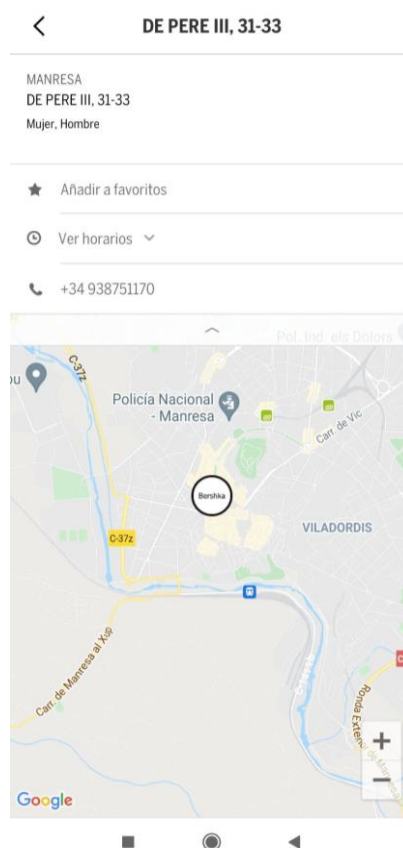


Fig. 8: Mapa de tienda Bershka

Después del análisis, se comenzó a editar los objetos base que ofrece Flex, cambiando los tamaños, formas, bordes, colores y añadiendo nuevas propiedades Fig.9, Fig.10 (p.ej. para adaptarlo al tamaño de la pantalla hice un parámetro que se indicaba con “pequeña”, “mediana” o “grande”).



Fig. 10: Toggle Switch predeterminado

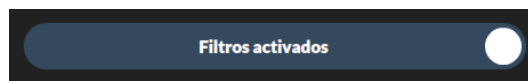


Fig. 9: Toggle Switch editado

Así que se utilizó el patrón de Modelo-vista-controlador Fig.11, donde el usuario puede reciclar el código de cada componente, desde el controlador, puede detectar los cambios que se quieran realizar para cada componente y así mostrarlos en la vista.

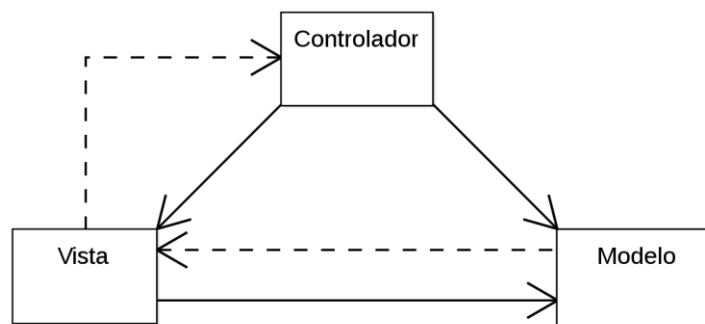


Fig. 11: MVC

Una vez modificados los componentes por defecto, se comenzó a editarlos, por colores, donde al dar un click, se pueda cambiar de tema de todos los objetos, así como el fondo de la app, se obtuvo de esa manera un tema oscuro y otro claro predeterminado.

Con el diseño listo, se empezó a realizar las funcionalidades que iba a tener, como mostrar artículos, crear usuarios, iniciar sesión, mostrar tiendas... Algunas de ellas no se pudieron comprobar si eran funcionales hasta el segundo semestre, donde se volvió a retomar el trabajo presencial, como tener los códigos de barras de los artículos y vales para poder comprobar si funcionaban necesitaba el escáner de la empresa y así testearlo.

A raíz de tener un proyecto más sólido, se comenzaron a realizar pruebas en físico desde un móvil Android, para comprobar que su funcionamiento seguía siendo correcto (funcionaban los botones, se podía desplegar el teclado, las listas se deslizaban...), que no se desmontasen los objetos por la pantalla, y para comprobarlo en diferentes pantallas se utilizó el simulador móvil de Eclipse.

Por la parte de servidor se usó FactoryBean Fig.12, para poder crear diferentes objetos desde una clase y así tener ordenado el código.

Este patrón está compuesto por:

- I+”objeto”: son las interfaces de las clases concretas, tienen definidas las funciones que tendrán.
- ”objeto”+SAP: son clases concretas que utilizan su objeto base e implementa su interfaz.
- ApplicationContext: es un fichero XML que contiene todos los objetos que deberán existir en la aplicación al inicializarse.
- beanFactory: es la fábrica que se dedicará a crear el tipo de objetos, en este caso SAP.
- ReplicaBase: contiene getters y setters de las interfaces de objetos, para poder utilizarla en las demás clases y hacer las llamadas a las funciones.

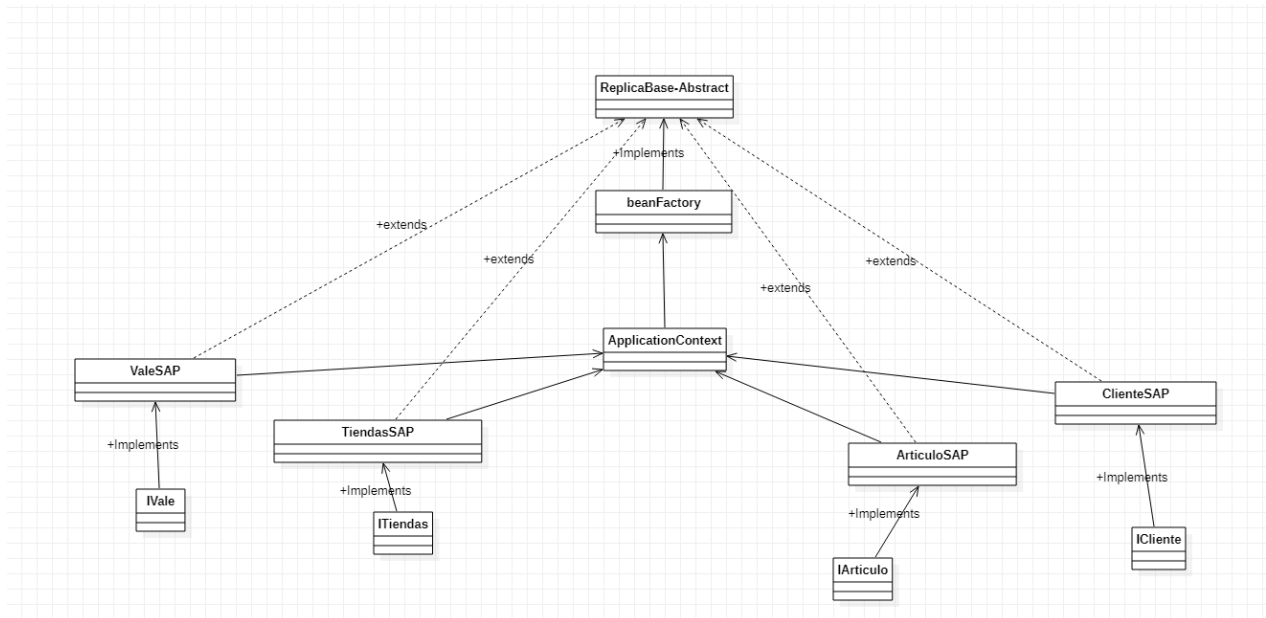


Fig. 12: Factory Bean diagrama UML

Otro patrón utilizado en varias ocasiones durante el desarrollo del proyecto fue el Singleton Fig.13, que se podía combinar entre las diferentes clases. Por ejemplo, para la variable global de la APP que guardaba el estado de ella, gracias a este patrón tan solo tenemos una instancia, así que utilizaremos siempre la misma variable.

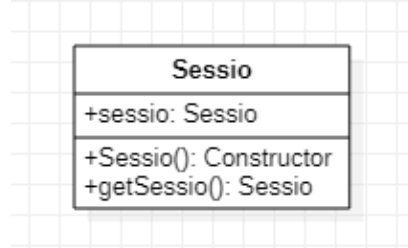


Fig. 13: Patrón Singleton UML

Implementación de la parte de BDD

Por la parte de la base de datos Fig.14, se ha utilizado la que tenía la empresa en SAP con la información necesaria para realizar una e Shop, ha sido organizada en varias partes, la primera es la parte del usuario, donde se guardan los datos personales de los clientes de las diferentes tiendas, con la contraseña codificada usando un hash en base 64 para aumentar la seguridad de los datos en caso de ataque. También existen los vales que tienen una relación de n... a 1 con el usuario, hay una relación con los vales que ha adquirido el cliente a partir de sus compras realizadas, este tendrá la información del número de vale para identificarlo, importe, fechas de caducidad y emisión, un código de barras para escanearlo desde la tienda y su estado que podrá ser expirado, no expirado o gastado. Por otro lado, está la relación de n... a 1 con la tienda, donde se ha obtenido por código de tienda, que será su clave primaria, esta tendrá información de contacto, de ubicación y una breve descripción, ya que si se quisiese más información se podrá encontrar mediante la conexión que tiene con Google Maps. En relación con las tiendas y artículos de la aplicación, tenemos una carpeta intermedia que guarda todas las imágenes nombradas por las claves primarias de sus objetos relacionados que acceden a ellas mediante las URLs, y tienen una relación de n a n. Los artículos tienen como clave primaria su código, de información contiene un código de barras, las tallas y colores donde la cantidad puede variar según el modelo y atributos relacionados con su precio.

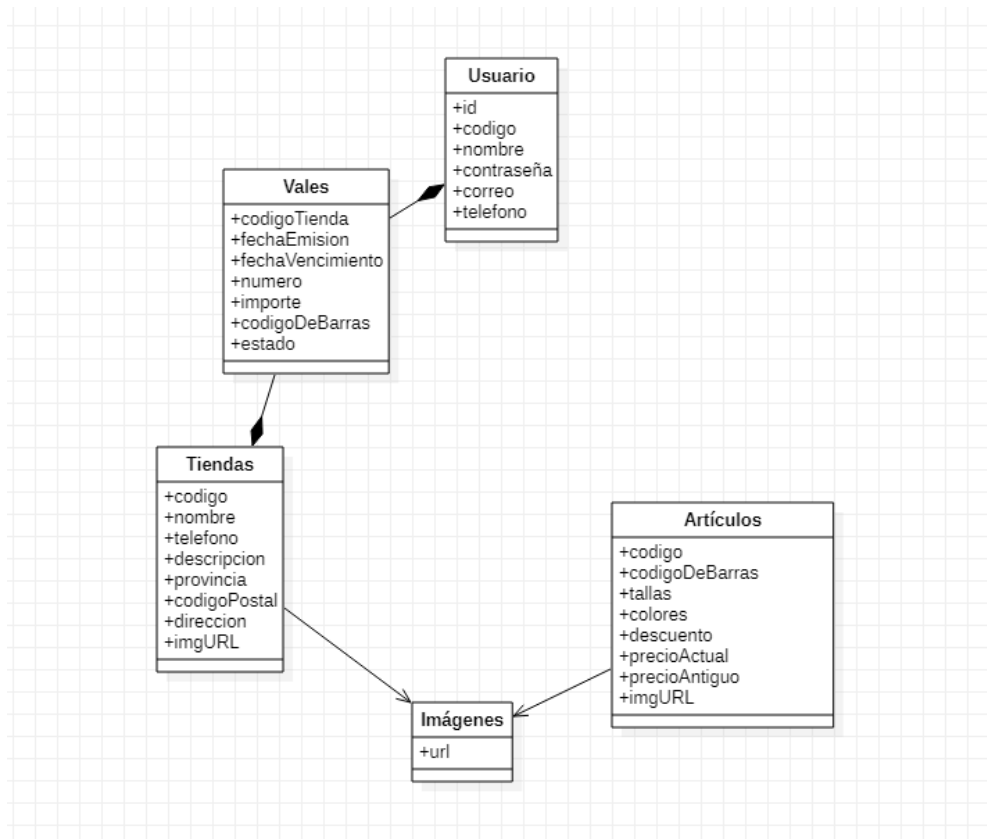


Fig. 14: Base de datos APP

La parte de servidor será iniciada como un servicio en uno de los servidores de la empresa, para que se ejecute desde cualquier dispositivo, este servicio es un Tomcat con la versión 7.0 que se instala en el servidor e implementa Java Servlet, un lenguaje de programación que sirve para ampliar las capacidades de un servidor. Este utilizará para comunicarse el protocolo HTTP, donde a través del puerto utilizado, se podrá comprobar si funciona a través de una página de pruebas, para probar diferentes conexiones que se puedan llamar durante el uso del servidor con el cliente. A partir de este punto y del FactoryBean comentado anteriormente se realizó una clase con las diferentes funciones que conectarían el servidor y el cliente, estas funciones podían devolver una lista de artículos, tiendas, vales, ..., un objeto concreto como cliente, artículo, vale..., comprobaciones que devuelven un boolean como comprobar que el usuario introducido para iniciar sesión es correcto. Desde la parte de cliente se utilizó un objeto de tipo RemoteObject (BEAN) Fig. 15 con el objetivo de que se realizar las diferentes consultas, donde gracias al objeto se puede encargar de hacer las peticiones correctamente y devolver el resultado que se necesite. En el ejemplo de Fig. 15 se llama a la función getArticles() si el resultado es correcto se guarda la lista en una variable, si no lo es, se llama a otra función que intenta recuperar la última lista descargada.

```

BEAN.getArticles().addEventListener("result", function correcte(event:ResultEvent):void{
    llista = event.result as ArrayCollection;
});
BEAN.getArticles().addEventListener("fault",function error (event:FaultEvent):void{
    getArticles();
});
BEAN.getArticles();

```

Fig. 15: Obtener artículos con RemoteObject

6. Resultados

La aplicación ha cumplido con la mayoría de los requisitos, pero no se ha podido terminar porque en el camino haciendo se han encontrado otros requisitos han hecho ralentizar el proceso de realización que no se tuvieron en cuenta en un primer momento.

6.1 Funcionalidades y producto final.

1. Cargar con icono *Fig.16*

Para que el usuario no se quedase esperando sin saber si se está abriendo la app u otra, se creó una pantalla de carga con el logo de la empresa (futuramente será la que se cree con la app) donde se intenta conectar con el servidor y cargar el idioma.



Intarex

Fig. 16: Cargar APP

2. Artículos *Fig.17*

Este apartado será el primero que se le muestre al usuario, por eso en la parte superior tendrá un icono, título de la APP que se está utilizando y un icono de acceso directo al apartado de usuario. En la lista del catálogo, se encuentra un filtro, que dependiendo de la empresa que tenga la aplicación, podrá utilizar entre uno y tres filtros y cambiarlos de nombre a su gusto. La lista mostrará la imagen del producto, su nombre y precio.



Fig. 17: Artículos APP

Por otro lado, cuando se le da clic a uno de los artículos, se muestra información en detalle, mostrando el precio, las tallas disponibles de la pieza, las tiendas en que se encuentran, los colores, un botón que muestra el código de barras para que el usuario lo pueda encontrar con más facilidad si la tienda dispone de escáneres y por último un botón para ver más imágenes en un tamaño de mayor resolución. Fig.18



Fig. 18: Detalle artículo APP

3. Usuario Fig.19

En el apartado de usuario se encuentra la información básica del contacto, esta es el nombre en la parte superior, el teléfono, el email y un código único de usuario, existe un botón que lleva al apartado de los vales que haya adquirido el usuario en las diferentes tiendas.

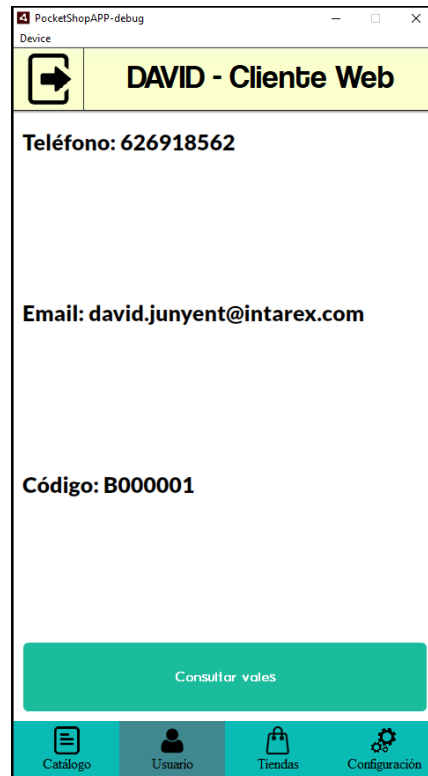


Fig. 19: Usuario APP

En la parte superior izquierda hay un botón de salida para desconectarse, donde se abre un pop-up, por si no ha sido su intención desconectarse y pueda escoger como segunda confirmación. Fig.20

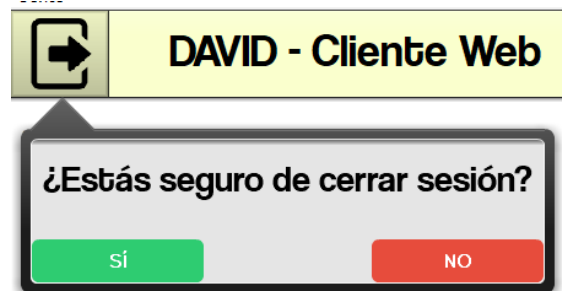


Fig. 20: Cerrar sesión APP

4. Iniciar sesión Fig.21

Si el usuario nunca ha iniciado sesión o se ha registrado, cuando le dé al botón de Usuario de la parte inferior o desde el apartado de artículos, se le redirigirá a la pantalla de Login, donde podrá escribir el nombre de usuario y la contraseña para iniciar sesión, si quiere registrarse será enviado a un formulario de una web donde pondrá la información personal básica (web aún no creada).

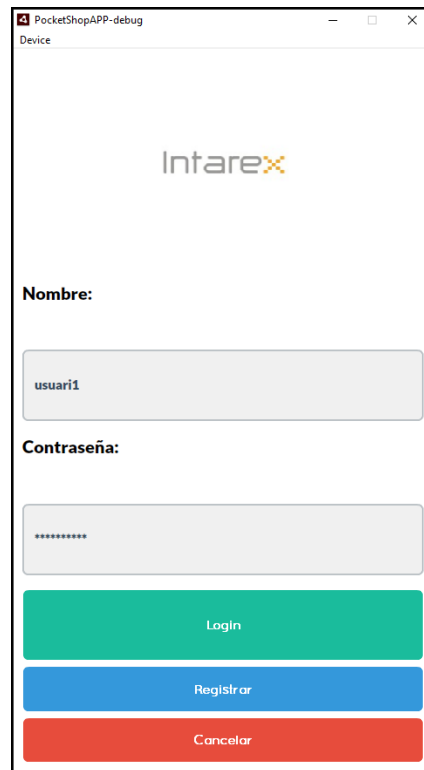


Fig. 21: Iniciar/Registrar usuario APP

5. Tiendas *Fig.22*

Este apartado se basa en un listado de las tiendas que tiene la empresa, mostrando información básica, una imagen de la tienda para reconocerla, el nombre y la localización de ella. Las imágenes de la base de datos se deberán añadir, ya que de momento no tienen ningún valor y la columna está a nulo.



Fig. 22: Tiendas APP

Si entramos al detalle de la tienda Fig.23, nos encontramos con la información dada anteriormente, el teléfono de contacto y con un botón de búsqueda que redirige a la tienda vista desde Google Maps Fig.24, si estás desde móvil te envía a la APP.



Fig. 23: Detalle tienda APP

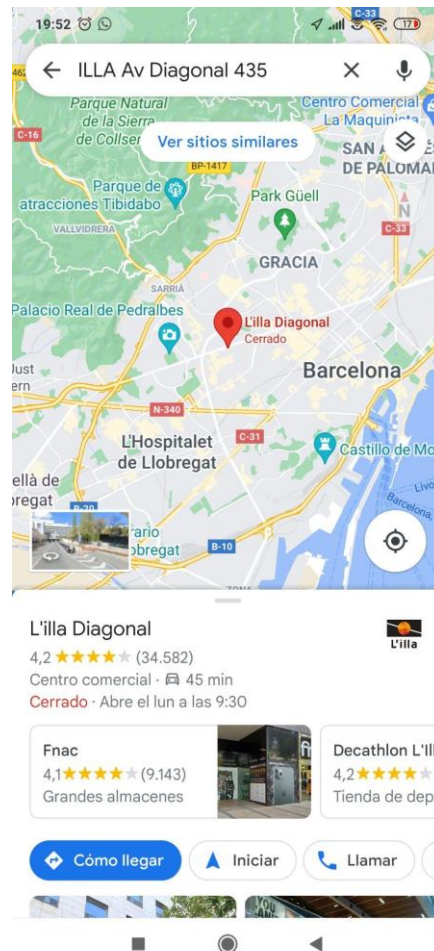


Fig. 24: Mapa de Google de Tienda

6. Configuración

Las opciones que proporciona el apartado de configuración son activar o desactivar filtros de los artículos, cambiar el tema a oscuro o claro, utilizando en ambos casos un “ToggleSwitch” (Interruptor de palanca). Para escoger el idioma se utilizaron los CheckBox(Casillas de verificación), de momento en español e inglés Fig.25. En el botón de “Más información”, se creó un pop-up donde se comprueban varias conexiones con el servidor para capturar lo que serían errores “típicos” que se puedan producir, esto podrá ayudar a que, si el cliente se pone en contacto con la empresa por problemas que pueda tener, se puedan resolver o detectar con facilidad. Fig.26



Fig. 25: Configuración APP



Fig. 26: Errores APP

7. Conclusiones

Durante la realización de este proyecto se aprendió cómo se trabaja en una aplicación de un proyecto real, conseguir realizar el proyecto con una alta autonomía, mejorar la capacidad de organizar un proyecto y llevarlo a

cabo, entender más en profundidad cómo funciona la empresa y qué recursos utilizan para conseguir una adaptación a ella.

No se pudo finalizar la aplicación para sacarla al mercado dado a la prioridad que le dio la empresa a otros proyectos por la acumulación de trabajo donde el estudiante tuvo que participar y dejar la aplicación de lado hasta que se termine esta temporada de trabajos. Así que el objetivo de integrar la app en el workflow de la empresa y poder analizar su impacto de su uso no pudo ser realizado. Y al realizar primero la versión de Android, por los problemas que da al crearla en iOS por sus estrictos requisitos y condiciones, la aplicación por el momento no es multiplataforma. En cambio, si se pudo lograr desarrollar una aplicación en flex, realizar diferentes pruebas con ella, crear un servidor mientras se aprendía a utilizar la API de SAP para entender como estaba estructurada la base de datos.

Un punto difícil de analizar ha sido conocer el entorno de trabajo y saber cómo trabajaban, ya que la mayoría de las horas que se han hecho han sido desde casa debido a la pandemia y esto complica mucho este tipo de aprendizaje. Otro punto ha sido poder comunicarse con los trabajadores de la empresa para realizar test a la APP, conseguir opiniones... Cosa que se ha cumplido con satisfacción.

Por otro lado, por el feedback recibido por parte de la empresa se cree que han salido satisfechos con el trabajo realizado, ya que, gracias a la oportunidad dada, se pudieron demostrar las capacidades del alumno y no hubo problemas con su rendimiento.

8. Bibliografía

- [1] Eclipse desktop & web IDEs: <https://www.eclipse.org/ide/>
- [2] Apache Flex: <http://flex.apache.org/>
- [3] MXML: <https://es.mihalicdictionary.org/wiki/MXML>
- [4] ActionScript: <https://sites.google.com/site/devaprepa2030/flash/actionscript-3-0>
- [5] SAP: https://www.sap.com/spain/why-sap.html?campaigncode=CRM-ES21-PPC-NSPBRAB&source=ppc-sp-GOO-71700000082515907-58700007002967705-sp-brand-fc-na&DFA=1&gclid=Cj0KCQjwl_SHBhCQARIsAFIFRVVXtQBA6xk9euUjdjJgZ3RHm4KHnzVMmFzHa5S5aAaxdcSzLP59KhsaAi4EEALw_wcB&gclsrc=aw.ds
- [6] SAP HANA: <https://www.sap.com/spain/products/hana.html>